Efficient and accurate modeling of rigid rods

Yongjun Pan[1] · Alfonso Callejo[2] · José L. Bueno[1] · Roger A. Wehage[3] · Javier García de Jalón[1]

Abstract Ten years ago, an original semi-recursive formulation for the dynamic simulation of large-scale multibody systems was presented by García de Jalón et al. (Advances in Computational Multibody Systems, pp. 1–23, 2005). By taking advantage of the cut-joint and rod-removal techniques through a double-step velocity transformation, this formulation proved to be remarkably efficient. The rod-removal technique was employed, primarily, to reduce the number of differential and constraint equations. As a result, inertia and external forces were applied to neighboring bodies. Those inertia forces depended on unknown accelerations, a fact that contributed to the complexity of the system inertia matrix. In search of performance improvement, this paper presents an approximation of rod-related inertia forces by using accelerations from previous time-steps. Additionally, a mass matrix partition is carried out to preserve the accuracy of the original formulation. Three extrapolation methods, namely, point, linear Lagrange and quadratic Lagrange extrapolation methods, are introduced to evaluate the unknown rod-related inertia forces. In order to assess the computational efficiency and solution accuracy of the presented approach, a general-purpose MATLAB/C/C++ simulation code is implemented. A 15-DOF, 12-rod sedan vehicle model with MacPherson strut and multi-link suspension systems is modeled, simulated and analyzed.

Keywords Multibody dynamics · Semi-recursive Maggi's formulation · Rod-removal technique · Polynomial extrapolation · Vehicle dynamics

1 Introduction

Vehicle dynamics has been a relevant application of multibody systems (MBS) since the 1970s. Standard road vehicles are very often considered as complex, closed-chain MBS

✉ Y. Pan
yongjun.pan@alumnos.upm.es

1 Computational Mechanics Group, INSIA, Universidad Politécnica de Madrid, Madrid 28031, Spain

2 Centre for Intelligent Machines, McGill University, Montreal, Quebec H3A 0E9, Canada

3 Caterpillar Inc., Mossville, IL, USA

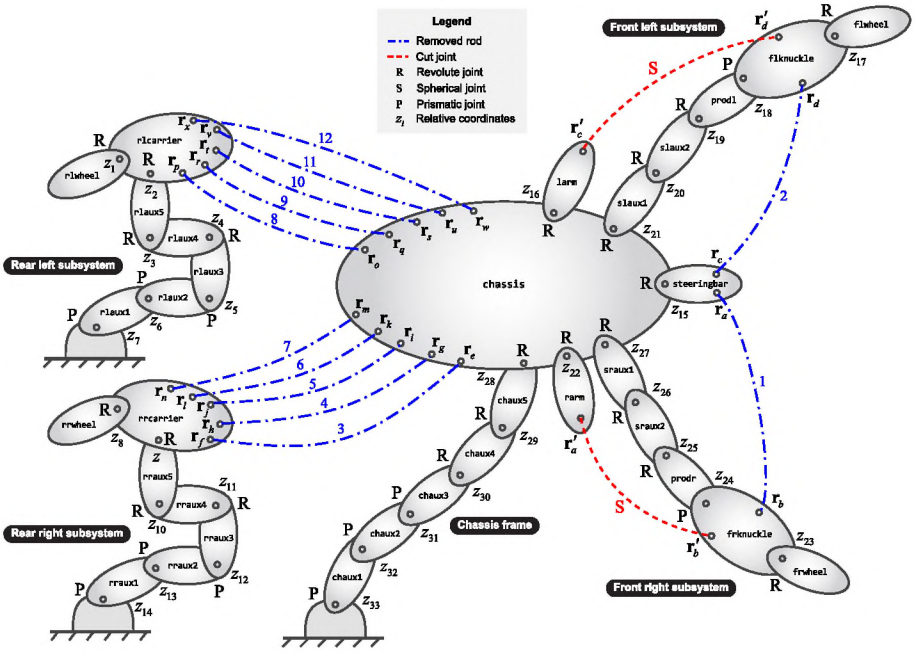

Fig. 1 System topology of the 15-DOF sedan vehicle model

with at least 15 degrees of freedom (DOF), namely: 6 DOF in the chassis frame (pitch, roll and yaw angles as well as X, Y and Z translations), 8 DOF in the front and rear suspension systems to describe the vertical suspension displacements and the rotation of the wheels, and 1 DOF to control the steering system. Figure 1 shows the tree-diagram topology of a sedan vehicle model with MacPherson strut suspension in the front axle and five-bar multi-link suspension in the rear axle. Several auxiliary massless bodies have been introduced to model spherical, universal and free joints by means of 1-DOF revolute and prismatic joints [10]. Further details about the vehicle model can be found in Appendix.

The vehicle model shown in Fig. 1 contains a number of rigid rods. A *rigid rod* (or simply *rod*) is hereby defined as a slender rigid body attached to the neighboring bodies through two spherical joints. It has a negligible moment of inertia around its centerline and a uniformly distributed mass. It can therefore be considered as a virtual, mathematical representation of a slender rigid link. In this context, it can be represented by using a constant-distance constraint and by applying its inertia properties—including coupling terms—to the neighboring bodies.

Closed loops in the MBS have been temporarily opened by eliminating all rods and "cutting" certain spherical joints. Once the closed-loops have been opened, recursion techniques, together with the connectivity information, can be employed to formulate the open-loop equations of motion in a simple and efficient manner. This step involves a *first velocity transformation* between Cartesian and joint coordinates that was proposed by Jerkovsky [15]. Then, by imposing a minimal set of loop-closure constraint equations through Lagrange multipliers, the closed-loop equations of motion can be obtained [22]. The numerical solution and integration requirements for this constraint enforcement approach are somewhat involved, and stabilization and computational efficiency are still challenging issues.

Cuadrado et al. [6–8] eliminated rods and proposed a formulation based on penalty terms representing the constraint forces generated from the cut joints. By using the augmented Lagrangian technique to avoid numerical ill-conditioning, accurate solutions were obtained. Other approaches for the improvement of computational efficiency are based on the advances of parallelization technology [1, 2, 5, 17, 18]. García de Jalón et al. [10, 12, 20] developed an alternative formulation of the equations of motion in terms of independent coordinates. By introducing a *second velocity transformation*, a set of independent relative accelerations was extracted from the dependent ones, which then allowed one to integrate the equations of motion using Maggi's equations in a fairly efficient way.

The second velocity transformation is constructed in terms of the constraint equations corresponding to opening the closed loops, which are first expressed in *natural* or *fully Cartesian coordinates*[1] and then converted to relative coordinates. After the transformation, the state-space equations of motion of the entire system are expressed using a minimum set of independent coordinates. Regarding the removal or opening of rod loops, instead of cutting one of their end spherical joints (which equates to three constraint equations) and replacing the other one by a universal joint, rods are introduced as just one constant-distance constraint equation. This decreases the number of relative coordinates and constraint equations. This algorithm constitutes a general-purpose formulation that has been effectively applied in numerous MBS applications [10, 12].

In this paper, we present an improvement to the aforementioned independent-coordinate formulation. When a rod is eliminated in the context of the rod-removal technique, its dynamic properties, including second-derivative-based inertia forces (SDIFs), velocity-dependent inertia forces (VDIFs) and external forces, can be exactly considered. Both rod-related VDIFs and external forces can be directly added to the connecting bodies; this cannot be done, however, with rod-related SDIFs because they are related to the unknown Cartesian accelerations. Usually, one combines this term with the independent accelerations to solve the ordinary differential equations. As a consequence, the leading mass matrix no longer has a simple form for efficient computation. If, on the other hand, this unknown term is approximated by a known one and added to the (known) right-hand side forces, the mass matrix remains in simple form, and the computational efficiency is improved. Furthermore, the rod mass matrix can be partitioned into a diagonal matrix and a coupling matrix. The diagonal part is kept on the left-hand side of the equations of motion and the coupling matrix is moved to the right-hand side, which improves the solution accuracy without any loss in computational efficiency.

Together with the analysis of rod-related inertia forces and the methods to approximate them, the 15-DOF sedan car in Fig. 1 is simulated to validate the algorithms and extract practical conclusions. Both accuracy and efficiency are assessed in the context of a general-purpose, MATLAB/C/C++ implementation.

2 Semi-recursive dynamic formulation

The connection between rigid bodies through revolute (R) and prismatic (P) joints can be modeled as shown in Fig. 2. Other kinds of kinematic joints with more than one DOF can be broken down into a series of massless auxiliary bodies connected by basic 1-DOF joints.

In order to pose the dynamic equations in a simple way, we choose as a reference the point $\mathbf{s}$ of body i that instantaneously coincides with the origin of the inertial reference

[1] Cartesian coordinates of points and Cartesian components of unit vectors [9, 11, 13].

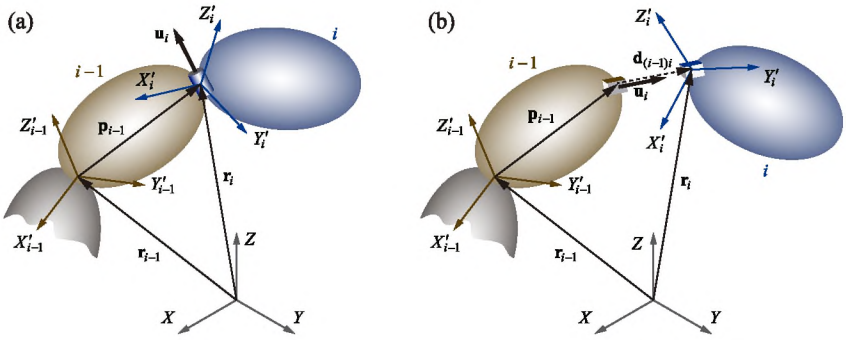

Fig. 2 (**a**) Revolute and (**b**) prismatic joints

frame, and we define the Cartesian velocities and accelerations of body i ($\mathbf{Z}_i$ and $\dot{\mathbf{Z}}_i$) and of the entire system ($\mathbf{Z}$ and $\dot{\mathbf{Z}}$) as:

$$\mathbf{Z}_i \equiv \begin{Bmatrix} \dot{\mathbf{s}}_i \\ \boldsymbol{\omega}_i \end{Bmatrix}, \qquad \mathbf{Z}^{\mathrm{T}} \equiv \left\{ \mathbf{Z}_1^{\mathrm{T}}, \mathbf{Z}_2^{\mathrm{T}}, \ldots, \mathbf{Z}_n^{\mathrm{T}} \right\}, \tag{1}$$

$$\dot{\mathbf{Z}}_i \equiv \begin{Bmatrix} \ddot{\mathbf{s}}_i \\ \dot{\boldsymbol{\omega}}_i \end{Bmatrix}, \qquad \dot{\mathbf{Z}}^{\mathrm{T}} \equiv \left\{ \dot{\mathbf{Z}}_1^{\mathrm{T}}, \dot{\mathbf{Z}}_2^{\mathrm{T}}, \ldots, \dot{\mathbf{Z}}_n^{\mathrm{T}} \right\} \tag{2}$$

where n is the number of bodies and $\boldsymbol{\omega}$ is the angular velocity. These coordinates have already been used by other authors [16, 19, 25].

In an open-loop system, the Cartesian velocities and accelerations can be recursively derived in a forward fashion on the basis of relative velocities and accelerations corresponding to joint coordinates:

$$\mathbf{Z}_i = \mathbf{Z}_{i-1} + \mathbf{b}_i \dot{z}_i, \tag{3}$$

$$\dot{\mathbf{Z}}_i = \dot{\mathbf{Z}}_{i-1} + \mathbf{b}_i \ddot{z}_i + \mathbf{d}_i \tag{4}$$

where $\dot{z}_i$ and $\ddot{z}_i$ are the relative velocities and accelerations of the ith joint, and vectors $\mathbf{b}_i$ and $\mathbf{d}_i$ can be calculated through expressions that depend on the type of the ith joint [10, 21, 26].

By combining Eqs. (1) and (3), we can introduce the following velocity transformation:

$$\mathbf{Z} = \mathbf{R}_1 \dot{z}_1 + \mathbf{R}_2 \dot{z}_2 + \cdots + \mathbf{R}_n \dot{z}_n = \mathbf{R}\dot{\mathbf{z}} \tag{5}$$

where matrix $\mathbf{R}$ allows one to express the Cartesian velocities in terms of relative velocities. This constitutes the first velocity transformation, which is essential for recursively calculating the open-loop velocities. The jth column of matrix $\mathbf{R}$ can be computed recursively because its elements correspond to Cartesian velocities of the bodies that are upwards in the topology tree, originated by a unit relative velocity in joint j and null relative velocities in the remaining ones [10]. Note that matrix $\mathbf{R}$ need not be calculated explicitly in the computer code.

Let us consider, as an example, the topology of the vehicle model presented in Fig. 1. Bodies have been numbered from the leaves to the root, so that children's numbers are less than their parents', as suggested by Negrut, Serban and Bae [3, 19, 24]. This prevents matrix

fill-in in further steps. Bodies have the same identification numbers as their input joints (one per body). According to the system topology defined above, the first velocity transformation matrix in Eq. (5) can be written as follows:

$$\mathbf{R} = \begin{bmatrix} \mathbf{B}_1^{\Sigma} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{B}_2^{\Sigma} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & \mathbf{b}_{15} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{\mathrm{CS}}^{\Sigma} \\ \vdots & & & \mathbf{B}_3^{\Sigma} & \mathbf{0} & \mathbf{B}_{\mathrm{CF}}^{\Sigma} \\ & \iddots & & & \mathbf{B}_4^{\Sigma} & \mathbf{B}_{\mathrm{CF}}^{\Sigma} \\ \mathbf{0} & & \cdots & & & \mathbf{B}_0^{\Sigma} \end{bmatrix} \equiv \mathbf{T}\mathbf{R}_{\mathrm{d}}, \tag{6}$$

$$\mathbf{B}_0^{\Sigma} \equiv \begin{bmatrix} \mathbf{b}_{28} & \mathbf{b}_{28} & \mathbf{b}_{28} & \mathbf{b}_{28} & \mathbf{b}_{28} & \mathbf{b}_{28} \\ & \mathbf{b}_{29} & \mathbf{b}_{29} & \mathbf{b}_{29} & \mathbf{b}_{29} & \mathbf{b}_{29} \\ & & \mathbf{b}_{30} & \mathbf{b}_{30} & \mathbf{b}_{30} & \mathbf{b}_{30} \\ \vdots & & & \mathbf{b}_{31} & \mathbf{b}_{31} & \mathbf{b}_{31} \\ & \iddots & & & \mathbf{b}_{32} & \mathbf{b}_{32} \\ \mathbf{0} & & \cdots & & & \mathbf{b}_{33} \end{bmatrix},$$

$$\mathbf{B}_1^{\Sigma} \equiv \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_1 & \mathbf{b}_1 & \mathbf{b}_1 & \mathbf{b}_1 & \mathbf{b}_1 & \mathbf{b}_1 \\ & \mathbf{b}_2 & \mathbf{b}_2 & \mathbf{b}_2 & \mathbf{b}_2 & \mathbf{b}_2 & \mathbf{b}_2 \\ & & \mathbf{b}_3 & \mathbf{b}_3 & \mathbf{b}_3 & \mathbf{b}_3 & \mathbf{b}_3 \\ \vdots & & & \mathbf{b}_4 & \mathbf{b}_4 & \mathbf{b}_4 & \mathbf{b}_4 \\ & \iddots & & & \mathbf{b}_5 & \mathbf{b}_5 & \mathbf{b}_5 \\ & & & & & \mathbf{b}_6 & \mathbf{b}_6 \\ \mathbf{0} & & & \cdots & & & \mathbf{b}_7 \end{bmatrix}, \tag{7}$$

$$\mathbf{B}_2^{\Sigma} \equiv \begin{bmatrix} \mathbf{b}_8 & \mathbf{b}_8 & \mathbf{b}_8 & \mathbf{b}_8 & \mathbf{b}_8 & \mathbf{b}_8 & \mathbf{b}_8 \\ & \mathbf{b}_9 & \mathbf{b}_9 & \mathbf{b}_9 & \mathbf{b}_9 & \mathbf{b}_9 & \mathbf{b}_9 \\ & & \mathbf{b}_{10} & \mathbf{b}_{10} & \mathbf{b}_{10} & \mathbf{b}_{10} & \mathbf{b}_{10} \\ \vdots & & & \mathbf{b}_{11} & \mathbf{b}_{11} & \mathbf{b}_{11} & \mathbf{b}_{11} \\ & \iddots & & & \mathbf{b}_{12} & \mathbf{b}_{12} & \mathbf{b}_{12} \\ & & & & & \mathbf{b}_{13} & \mathbf{b}_{13} \\ \mathbf{0} & & & \cdots & & & \mathbf{b}_{14} \end{bmatrix},$$

$$\mathbf{B}_3^{\Sigma} \equiv \begin{bmatrix} \mathbf{b}_{16} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{b}_{17} & \mathbf{b}_{17} & \mathbf{b}_{17} & \mathbf{b}_{17} & \mathbf{b}_{17} \\ & & \mathbf{b}_{18} & \mathbf{b}_{18} & \mathbf{b}_{18} & \mathbf{b}_{18} \\ \vdots & & & \mathbf{b}_{19} & \mathbf{b}_{19} & \mathbf{b}_{19} \\ & \iddots & & & \mathbf{b}_{20} & \mathbf{b}_{20} \\ \mathbf{0} & & \cdots & & & \mathbf{b}_{21} \end{bmatrix}, \tag{8}$$

$$\mathbf{B}_4^{\Sigma} \equiv \begin{bmatrix} \mathbf{b}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{b}_{23} & \mathbf{b}_{23} & \mathbf{b}_{23} & \mathbf{b}_{23} & \mathbf{b}_{23} \\ & & \mathbf{b}_{24} & \mathbf{b}_{24} & \mathbf{b}_{24} & \mathbf{b}_{24} \\ \vdots & & & \mathbf{b}_{25} & \mathbf{b}_{25} & \mathbf{b}_{25} \\ & \iddots & & & \mathbf{b}_{26} & \mathbf{b}_{26} \\ \mathbf{0} & & \cdots & & & \mathbf{b}_{27} \end{bmatrix},$$

$$\mathbf{B}_{\mathrm{CS}}^{\Sigma} \equiv \begin{bmatrix} \mathbf{b}_{28} & \mathbf{b}_{29} & \mathbf{b}_{30} & \mathbf{b}_{31} & \mathbf{b}_{32} & \mathbf{b}_{33} \end{bmatrix},$$

$$\mathbf{B}_{\mathrm{CF}}^{\Sigma} \equiv \begin{bmatrix} \mathbf{b}_{28} & \mathbf{b}_{29} & \mathbf{b}_{30} & \mathbf{b}_{31} & \mathbf{b}_{32} & \mathbf{b}_{33} \\ \mathbf{b}_{28} & \mathbf{b}_{29} & \mathbf{b}_{30} & \mathbf{b}_{31} & \mathbf{b}_{32} & \mathbf{b}_{33} \\ \mathbf{b}_{28} & \mathbf{b}_{29} & \mathbf{b}_{30} & \mathbf{b}_{31} & \mathbf{b}_{32} & \mathbf{b}_{33} \\ \mathbf{b}_{28} & \mathbf{b}_{29} & \mathbf{b}_{30} & \mathbf{b}_{31} & \mathbf{b}_{32} & \mathbf{b}_{33} \\ \mathbf{b}_{28} & \mathbf{b}_{29} & \mathbf{b}_{30} & \mathbf{b}_{31} & \mathbf{b}_{32} & \mathbf{b}_{33} \\ \mathbf{b}_{28} & \mathbf{b}_{29} & \mathbf{b}_{30} & \mathbf{b}_{31} & \mathbf{b}_{32} & \mathbf{b}_{33} \end{bmatrix} \tag{9}$$

where matrix $\mathbf{T}$ is the path matrix, which represents the connectivity of the multibody system, and $\mathbf{R}_{\mathrm{d}}$ is a diagonal matrix whose elements are the $\mathbf{b}_i$ vectors defined in Eq. (3). Submatrices $\mathbf{B}_1^{\Sigma}$, $\mathbf{B}_2^{\Sigma}$, $\mathbf{B}_3^{\Sigma}$, $\mathbf{B}_4^{\Sigma}$, and $\mathbf{B}_0^{\Sigma}$ are the elements of the first velocity transformation matrix corresponding to the rear left, rear right, front left and front right suspension systems and the chassis frame, respectively. Submatrices $\mathbf{B}_{\mathrm{CS}}^{\Sigma}$ and $\mathbf{B}_{\mathrm{CF}}^{\Sigma}$ correspond to the coupling between the chassis and the steering and front suspension systems, respectively.

The n equations of motion of the open-chain MBS can then be obtained by applying the method of virtual power and introducing the first velocity transformation ($\mathbf{R} \equiv \mathbf{T}\mathbf{R}_{\mathrm{d}}$) according to Jerkovsky and Schwerin [15, 22]:

$$\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\mathbf{T}\mathbf{R}_{\mathrm{d}}\ddot{\mathbf{z}} = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}(\bar{\mathbf{Q}} - \bar{\mathbf{P}}) \tag{10}$$

or, in compact notation,

$$\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\ddot{\mathbf{z}} = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left(\mathbf{Q}^{\Sigma} - \mathbf{P}^{\Sigma}\right) \tag{11}$$

where $\bar{\mathbf{M}}$, $\bar{\mathbf{Q}}$ and $\bar{\mathbf{P}}$ respectively are the global inertia matrix, external force vector and velocity-dependent inertia forces (VDIFs) expressed with respect to the aforementioned reference point. The matrices $\mathbf{M}^{\Sigma}$, $\mathbf{Q}^{\Sigma}$ and $\mathbf{P}^{\Sigma}$ contain their corresponding accumulated terms [10].

By means of a second velocity transformation, $\mathbf{R}_{\mathrm{z}}$, the relative velocities ($\dot{\mathbf{z}}$) and accelerations ($\ddot{\mathbf{z}}$) can be expressed in terms of independent relative velocities ($\dot{\mathbf{z}}^{\mathrm{i}}$) and accelerations ($\ddot{\mathbf{z}}^{\mathrm{i}}$):

$$\dot{\mathbf{z}} = \mathbf{R}_{\mathrm{z}}\dot{\mathbf{z}}^{\mathrm{i}}, \tag{12}$$

$$\ddot{\mathbf{z}} = \mathbf{R}_{\mathrm{z}}\ddot{\mathbf{z}}^{\mathrm{i}} + \dot{\mathbf{R}}_{\mathrm{z}}\dot{\mathbf{z}}^{\mathrm{i}}. \tag{13}$$

This approach corresponds to a state-space formulation of the equations of motion via the *embedding* or *matrix-R method* [10, 12, 20]. Introducing these expressions into the first velocity transformation (Eq. (5)), we also obtain:

$$\mathbf{Z} = \mathbf{R}\dot{\mathbf{z}} = \mathbf{T}\mathbf{R}_{\mathrm{d}}\dot{\mathbf{z}} = \mathbf{T}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}}\dot{\mathbf{z}}^{\mathrm{i}}, \tag{14}$$

$$\dot{\mathbf{Z}} = \mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} = \mathbf{T}\mathbf{R}_{\mathrm{d}}\ddot{\mathbf{z}} + \mathbf{T}\dot{\mathbf{R}}_{\mathrm{d}}\dot{\mathbf{z}} = \mathbf{T}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}}\ddot{\mathbf{z}}^{\mathrm{i}} + \mathbf{T}\frac{d(\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}})}{dt}\dot{\mathbf{z}}^{\mathrm{i}} \tag{15}$$

where the following equivalence, which is derived in the literature [12], holds:

$$\dot{\mathbf{R}}_{\mathrm{d}}\dot{\mathbf{z}} + \mathbf{R}_{\mathrm{d}}\dot{\mathbf{R}}_{\mathrm{z}}\dot{\mathbf{z}}^{\mathrm{i}} = \frac{d(\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}})}{dt}\dot{\mathbf{z}}^{\mathrm{i}}. \tag{16}$$

Matrix $\mathbf{R}_{\mathrm{z}}$ can be computed by formulating the Jacobian matrix of the m loop-closure constraint equations and generating a basis of the Jacobian matrix nullspace. Figure 3 depicts the opening of closed chains by removing a rod and by cutting a spherical and a revolute joint. The corresponding loop-closure (holonomic) constraint equations can be written

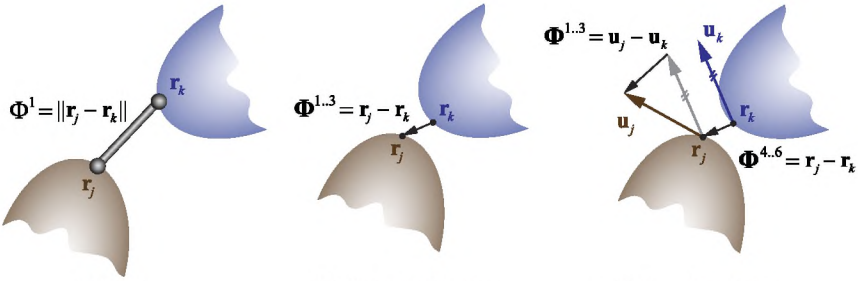

Fig. 3 Rod, spherical joint and revolute joint constraint equations; note that only 5 out of 6 equations are independent in the revolute joint case

in terms of natural coordinates, that is, points ($\mathbf{r}$) and unit vectors ($\mathbf{u}$) that belong to the bodies at hand. The constraints are then gathered in a constraint vector $\mathbf{\Phi}(\mathbf{q})$ and, when needed, expressed in terms of relative coordinates (i.e., $\mathbf{\Phi}(\mathbf{z})$) by using the chain rule of differentiation [12]. See [10] for more details.

Taking into account the loop-closure constraint equations and introducing the second velocity transformation ($\mathbf{R}_z$) in Eq. (10), in the form of Eqs. (13) and (16), the final set of ordinary differential equations (ODE) in independent coordinates is obtained:

$$\mathbf{R}_z^{\mathrm{T}}\mathbf{R}_d^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_d\mathbf{R}_z\ddot{\mathbf{z}}^{\mathrm{i}} = \mathbf{R}_z^{\mathrm{T}}\mathbf{R}_d^{\mathrm{T}}\left[\mathbf{Q}^{\Sigma} - \mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\frac{d(\mathbf{T}\mathbf{R}_d\mathbf{R}_z)}{dt}\dot{\mathbf{z}}^{\mathrm{i}}\right] \tag{17}$$

where there is one dynamic equation per DOF, that is, $f = n - m$ equations. The detailed expressions for $\mathbf{R}_z$, $\bar{\mathbf{M}}$, $\bar{\mathbf{Q}}$, $\bar{\mathbf{P}}$, $\mathbf{M}^{\Sigma}$, $\mathbf{Q}^{\Sigma}$ and $\mathbf{P}^{\Sigma}$ in Eqs. (10) and (17) can be found in [10]. Only holonomic and scleronomic constraint equations (but possibly redundant, as long as they are compatible) are considered here. The selection of independent coordinates is assumed to be valid throughout the entire simulation, which is typically the case in vehicle dynamics [23].

3 Rod-removal technique

Rods are very often part of vehicle suspension systems. They not only constitute a numerical burden because of their numerous joint coordinates and contribution to the body count, but they also present numerical challenges associated with the neglected rotation around their axis [10]. Not uncommonly, rods are only considered as constant-distance constraints (thus neglecting the inertia forces associated with them) or are forbidden (for example, requiring the replacement of one of the spherical joints by a universal joint). One of the main features of the presented formulation is the rod-removal technique and full consideration of their inertia properties [19], which makes it remarkably efficient and realistic. This technique, as already mentioned, is particularly attractive for vehicle models.

Applying the rod-removal technique to open the closed chains not only reduces the size of the dynamic equations, but also decreases the number of constraint equations. The reason is that removing one rod adds only one constant-distance constraint equation (recall Fig. 3) and, in turn, eliminates one rigid body and two spherical joints. On the other hand, a cut spherical joint implies three constraint equations with no elimination of bodies, and a cut

Table 1 Effect of cutting methods on constraints, bodies and relative coordinates

Loop-closure method	# Constraints	# Eliminated constraints	# Eliminated bodies	# Eliminated relative coordinates
Rod	1	2	1	6
Spherical joint	3	1	0	3
Revolute joint	5	1	0	1

revolute joint implies five constraint equations with no elimination of bodies. Table 1 summarizes the effect of the different loop-closure methods on constraint equations and relative coordinates.

Let us look at the inertia and external forces that must be added to the neighboring bodies. According to [10], a rod connected to bodies j and k produces the following Cartesian forces:

$$\begin{Bmatrix} \bar{\mathbf{Q}}_j \\ \bar{\mathbf{Q}}_k \end{Bmatrix} = -\begin{bmatrix} \bar{\mathbf{M}}_{jj} & \bar{\mathbf{M}}_{jk} \\ \bar{\mathbf{M}}_{kj} & \bar{\mathbf{M}}_{kk} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{Z}}_j \\ \dot{\mathbf{Z}}_k \end{Bmatrix} - \begin{Bmatrix} \begin{bmatrix} \mathbf{I}_3 \\ \tilde{\mathbf{r}}_j \end{bmatrix} \left(\frac{m}{3} \tilde{\boldsymbol{\omega}}_j \tilde{\boldsymbol{\omega}}_j \mathbf{r}_j + \frac{m}{6} \tilde{\boldsymbol{\omega}}_k \tilde{\boldsymbol{\omega}}_k \mathbf{r}_k \right) \\ \begin{bmatrix} \mathbf{I}_3 \\ \tilde{\mathbf{r}}_k \end{bmatrix} \left(\frac{m}{6} \tilde{\boldsymbol{\omega}}_j \tilde{\boldsymbol{\omega}}_j \mathbf{r}_j + \frac{m}{3} \tilde{\boldsymbol{\omega}}_k \tilde{\boldsymbol{\omega}}_k \mathbf{r}_k \right) \end{Bmatrix} + \begin{Bmatrix} \mathbf{f}_j \\ \tilde{\mathbf{r}}_j \mathbf{f}_j \\ \mathbf{f}_k \\ \tilde{\mathbf{r}}_k \mathbf{f}_k \end{Bmatrix}, \tag{18}$$

$$\bar{\mathbf{M}}_{jj} \equiv \frac{m}{3} \begin{bmatrix} \mathbf{I}_3 & \tilde{\mathbf{r}}_j^{\mathrm{T}} \\ \tilde{\mathbf{r}}_j & \tilde{\mathbf{r}}_j \tilde{\mathbf{r}}_j^{\mathrm{T}} \end{bmatrix}, \qquad \bar{\mathbf{M}}_{kk} \equiv \frac{m}{3} \begin{bmatrix} \mathbf{I}_3 & \tilde{\mathbf{r}}_k^{\mathrm{T}} \\ \tilde{\mathbf{r}}_k & \tilde{\mathbf{r}}_k \tilde{\mathbf{r}}_k^{\mathrm{T}} \end{bmatrix}, \qquad \bar{\mathbf{M}}_{jk} \equiv \bar{\mathbf{M}}_{kj}^{T} \equiv \frac{m}{6} \begin{bmatrix} \mathbf{I}_3 & \tilde{\mathbf{r}}_k^{\mathrm{T}} \\ \tilde{\mathbf{r}}_j & \tilde{\mathbf{r}}_j \tilde{\mathbf{r}}_k^{\mathrm{T}} \end{bmatrix} \tag{19}$$

where m is the mass of the rod, $\bar{\mathbf{Q}}_j$ and $\bar{\mathbf{Q}}_k$ are the forces propagated to the neighboring bodies through points j and k, and $\mathbf{f}_j$ and $\mathbf{f}_k$ are the external forces applied on points j and k of the rod. Note that these force terms are expressed with respect to the previously defined global reference point. The upper tilde represents the skew-symmetric matrix associated with the cross-product operator. Equation (18) is expressed more compactly as

$$\begin{Bmatrix} \bar{\mathbf{Q}}_j \\ \bar{\mathbf{Q}}_k \end{Bmatrix} = -\begin{bmatrix} \bar{\mathbf{M}}_{jj} & \bar{\mathbf{M}}_{jk} \\ \bar{\mathbf{M}}_{kj} & \bar{\mathbf{M}}_{kk} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{Z}}_j \\ \dot{\mathbf{Z}}_k \end{Bmatrix} - \begin{Bmatrix} \bar{\mathbf{P}}_j \\ \bar{\mathbf{P}}_k \end{Bmatrix}_{\text{rod}} + \begin{Bmatrix} \bar{\mathbf{Q}}_j \\ \bar{\mathbf{Q}}_k \end{Bmatrix}_{\text{rod}}. \tag{20}$$

The three terms on the right-hand side of Eqs. (18) and (20) are the rod-related second-derivative-based inertia forces (SDIFs), velocity-dependent inertia forces (VDIFs) and external forces. The rod-related VDIFs and external forces can be directly added to the corresponding neighboring body forces. On the other hand, the rod-related SDIFs are unknown, as they couple the neighboring Cartesian accelerations via the mass matrices of the rods. Usually, this term would be added to the left-hand side of the dynamic equations and solved with the rest of the unknown accelerations. Let us update the open-loop equations in Eq. (10) with the new terms:

$$\mathbf{R}_{\mathrm{d}}^{\mathrm{T}} \mathbf{T}^{\mathrm{T}} \bar{\mathbf{M}} \mathbf{T} \mathbf{R}_{\mathrm{d}} \ddot{\mathbf{z}} + \mathbf{R}_{\mathrm{d}}^{\mathrm{T}} \mathbf{T}^{\mathrm{T}} \bar{\mathbf{M}}_{\mathrm{rods}} \dot{\mathbf{Z}}_{\mathrm{rods}} = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}} \mathbf{T}^{\mathrm{T}} \left[(\bar{\mathbf{Q}} + \bar{\mathbf{Q}}_{\mathrm{rods}}) - (\bar{\mathbf{P}} + \bar{\mathbf{P}}_{\mathrm{rods}}) \right] \tag{21}$$

where $\bar{\mathbf{M}}_{\mathrm{rods}}$, $\bar{\mathbf{Q}}_{\mathrm{rods}}$ and $\bar{\mathbf{P}}_{\mathrm{rods}}$ are the contributions of the rods to the inertia matrix, the external forces and the VDIFs of the eliminated rods, respectively. In compact form,

$$\mathbf{R}_{\mathrm{d}}^{\mathrm{T}} \mathbf{M}^{\Sigma} \mathbf{R}_{\mathrm{d}} \ddot{\mathbf{z}} + \mathbf{R}_{\mathrm{d}}^{\mathrm{T}} \tilde{\mathbf{F}}_{\mathrm{rods}}^{\Sigma} = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}} \left(\mathbf{Q}^{\Sigma} - \mathbf{P}^{\Sigma} \right) + \mathbf{R}_{\mathrm{d}}^{\mathrm{T}} \left(\mathbf{Q}_{\mathrm{rods}}^{\Sigma} - \mathbf{P}_{\mathrm{rods}}^{\Sigma} \right) \tag{22}$$

Table 2 Contribution of rods to the global external and inertia forces

	Rod-related external forces ($\bar{\mathbf{Q}}_{\text{rods}}$)	Rod-related velocity-dependent inertia forces ($\bar{\mathbf{P}}_{\text{rods}}$)
Rear left knuckle	$\bar{\mathbf{Q}}_p + \bar{\mathbf{Q}}_r + \bar{\mathbf{Q}}_t + \bar{\mathbf{Q}}_v + \bar{\mathbf{Q}}_s$	$\bar{\mathbf{P}}_p + \bar{\mathbf{P}}_r + \bar{\mathbf{P}}_t + \bar{\mathbf{P}}_v + \bar{\mathbf{P}}_s$
Rear right knuckle	$\bar{\mathbf{Q}}_f + \bar{\mathbf{Q}}_h + \bar{\mathbf{Q}}_j + \bar{\mathbf{Q}}_l + \bar{\mathbf{Q}}_n$	$\bar{\mathbf{P}}_f + \bar{\mathbf{P}}_h + \bar{\mathbf{P}}_j + \bar{\mathbf{P}}_l + \bar{\mathbf{P}}_n$
Steering bar	$\bar{\mathbf{Q}}_a + \bar{\mathbf{Q}}_c$	$\bar{\mathbf{P}}_a + \bar{\mathbf{P}}_c$
Front left knuckle	$\bar{\mathbf{Q}}_d$	$\bar{\mathbf{P}}_d$
Front right knuckle	$\bar{\mathbf{Q}}_b$	$\bar{\mathbf{P}}_b$
Chassis frame	$\bar{\mathbf{Q}}_e + \bar{\mathbf{Q}}_g + \bar{\mathbf{Q}}_i + \bar{\mathbf{Q}}_k + \bar{\mathbf{Q}}_m + \bar{\mathbf{Q}}_o + \bar{\mathbf{Q}}_q + \bar{\mathbf{Q}}_s + \bar{\mathbf{Q}}_u + \bar{\mathbf{Q}}_w$	$\bar{\mathbf{P}}_e + \bar{\mathbf{P}}_g + \bar{\mathbf{P}}_i + \bar{\mathbf{P}}_k + \bar{\mathbf{P}}_m + \bar{\mathbf{P}}_o + \bar{\mathbf{P}}_q + \bar{\mathbf{P}}_s + \bar{\mathbf{P}}_u + \bar{\mathbf{P}}_w$

Table 3 Contribution of rods to the global inertia matrix ($\bar{\mathbf{M}}_{\text{rods}}$)

	Rear left knuckle	Rear right knuckle	Steering bar	Front left knuckle	Front right knuckle	Chassis frame
Rear left knuckle	$\bar{\mathbf{M}}_{pp} + \bar{\mathbf{M}}_{rr} + \bar{\mathbf{M}}_{tt} + \bar{\mathbf{M}}_{vv} + \bar{\mathbf{M}}_{xx}$					Sym.
Rear right knuckle		$\bar{\mathbf{M}}_{ff} + \bar{\mathbf{M}}_{hh} + \bar{\mathbf{M}}_{jj} + \bar{\mathbf{M}}_{ll} + \bar{\mathbf{M}}_{nn}$				
Steering bar			$\bar{\mathbf{M}}_{aa} + \bar{\mathbf{M}}_{cc}$			
Front left knuckle			$\bar{\mathbf{M}}_{cd}$	$\bar{\mathbf{M}}_{dd}$		
Front right knuckle			$\bar{\mathbf{M}}_{ab}$		$\bar{\mathbf{M}}_{bb}$	
Chassis frame	$\bar{\mathbf{M}}_{op} + \bar{\mathbf{M}}_{qr} + \bar{\mathbf{M}}_{st} + \bar{\mathbf{M}}_{uv} + \bar{\mathbf{M}}_{wx}$	$\bar{\mathbf{M}}_{ef} + \bar{\mathbf{M}}_{gh} + \bar{\mathbf{M}}_{ij} + \bar{\mathbf{M}}_{kl} + \bar{\mathbf{M}}_{mn}$				$\bar{\mathbf{M}}_{ee} + \bar{\mathbf{M}}_{gg} + \bar{\mathbf{M}}_{ii} + \bar{\mathbf{M}}_{kk} + \bar{\mathbf{M}}_{mm} + \bar{\mathbf{M}}_{oo} + \bar{\mathbf{M}}_{qq} + \bar{\mathbf{M}}_{ss} + \bar{\mathbf{M}}_{uu} + \bar{\mathbf{M}}_{ww}$

where $\tilde{\mathbf{F}}^{\Sigma}_{\text{rods}} \equiv \mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}_{\text{rods}}\dot{\mathbf{Z}}_{\text{rods}}$, $\mathbf{Q}^{\Sigma}_{\text{rods}} \equiv \mathbf{T}^{\mathrm{T}}\bar{\mathbf{Q}}_{\text{rods}}$ and $\mathbf{P}^{\Sigma}_{\text{rods}} \equiv \mathbf{T}^{\mathrm{T}}\bar{\mathbf{P}}_{\text{rods}}$ are the accumulated rod-related SDIFs, external forces and VDIFs due to the presence of rods. Tables 2 and 3 summarize these contributions for the case of the vehicle model.

By introducing Eq. (15) into Eq. (22), the rod-related SDIFs can be expressed in terms of relative accelerations. Then, by adding the corresponding rod-related VDIFs and external forces to the right-hand side of Eq. (22), we arrive at the following set of open-loop equations in relative coordinates:

$$\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left(\mathbf{M}^{\Sigma} + \mathbf{M}^{\Sigma}_{\text{rods}}\right)\mathbf{R}_{\mathrm{d}}\ddot{\mathbf{z}} = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left[\left(\mathbf{Q}^{\Sigma} + \mathbf{Q}^{\Sigma}_{\text{rods}}\right) - \left(\mathbf{P}^{\Sigma} + \mathbf{P}^{\Sigma}_{\text{rods}}\right) - \mathbf{M}^{\Sigma}_{\text{rods}}\dot{\mathbf{R}}_{\mathrm{d}}\dot{\mathbf{z}}\right] \tag{23}$$

where $\mathbf{M}^{\Sigma}_{\text{rods}} \equiv \mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}_{\text{rods}}\mathbf{T}$ is the accumulated inertia associated with the rods.

As already mentioned, by taking into account the loop-closure constraints, a second velocity transformation ($\mathbf{R}_z$) can be computed to express the relative coordinates in terms of

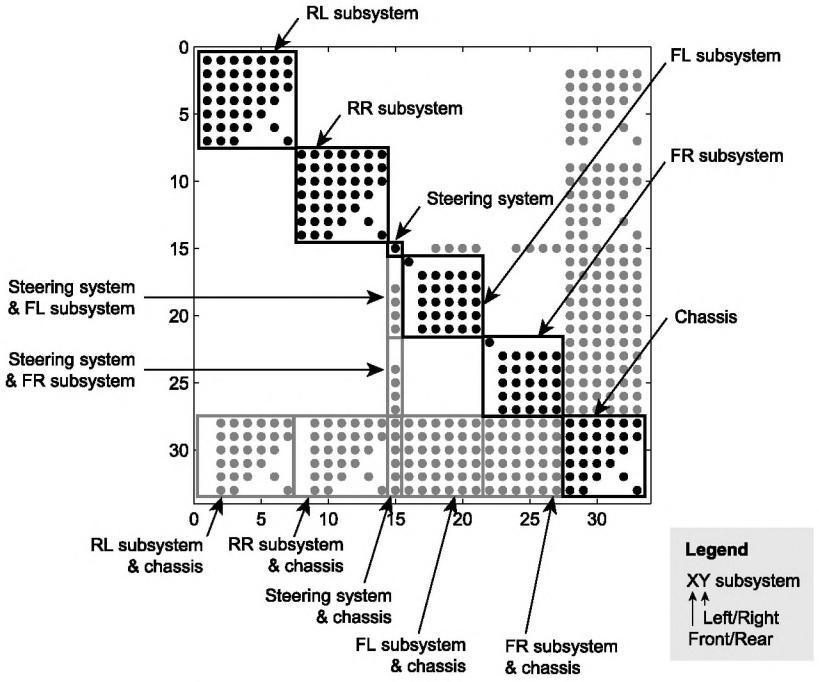

Fig. 4 Structure of matrix $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\hat{\mathbf{M}}^{\Sigma}\mathbf{R}_{\mathrm{d}}$ including rod terms

independent relative coordinates. We can then obtain the final set of motion differential equations in independent coordinates by introducing Eqs. (13) and (16):

$$\begin{aligned}&\mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\underbrace{\left(\mathbf{M}^{\Sigma}+\mathbf{M}_{\mathrm{rods}}^{\Sigma}\right)}_{\hat{\mathbf{M}}^{\Sigma}}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}}\ddot{\mathbf{z}}^{\mathrm{i}}\\&=\mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left[\underbrace{\left(\mathbf{Q}^{\Sigma}+\mathbf{Q}_{\mathrm{rods}}^{\Sigma}\right)}_{\hat{\mathbf{Q}}^{\Sigma}}-\mathbf{T}^{\mathrm{T}}\underbrace{\left(\bar{\mathbf{M}}+\bar{\mathbf{M}}_{\mathrm{rods}}\right)}_{\hat{\bar{\mathbf{M}}}}\frac{d\left(\mathbf{T}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}}\right)}{dt}\dot{\mathbf{z}}^{\mathrm{i}}-\mathbf{P}_{\mathrm{rods}}^{\Sigma}\right]\end{aligned}\tag{24}$$

where $\hat{\mathbf{M}}^{\Sigma}$, $\hat{\mathbf{Q}}^{\Sigma}$ and $\hat{\bar{\mathbf{M}}}$ are the accumulated inertia matrix, the system external forces and the inertia matrix including the rods' contribution, respectively.

4 Approximation of rod-related SDIFs

The inclusion of rod terms adds complexity to the inertia matrices by filling them with the coupling terms shown in Table 3, as can be seen by comparing Eqs. (17) and (24). We will now examine the two velocity transformations to identify diagonal and off-diagonal terms. Specifically, let us analyze the open-loop mass matrix $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\hat{\mathbf{M}}^{\Sigma}\mathbf{R}_{\mathrm{d}}$ in Eq. (24). The reason for not studying $\hat{\mathbf{M}}^{\Sigma}$ itself is that, for improved computational efficiency, we do not explicitly compute it in the computer code. Instead, we recursively compute $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\hat{\mathbf{M}}^{\Sigma}\mathbf{R}_{\mathrm{d}}$ in a direct way. Figure 4 shows the sparsity pattern of matrix $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\hat{\mathbf{M}}^{\Sigma}\mathbf{R}_{\mathrm{d}}$. The different squares and labels

show the diagonal and off-diagonal submatrices. This matrix constitutes a sparse, symmetric matrix, and has the following properties in the vehicle example:

- The four suspension systems are independent of each other.
- The steering and rear suspension systems are independent of each other.
- The steering and front suspension systems are coupled.
- The chassis frame is coupled with the suspension and steering systems.
- Each front and rear suspension subsystem is associated with six and seven relative coordinates, respectively.
- The steering system coordinate is known (predefined) and located at the center of the relative coordinate vector.
- There are six relative coordinates associated with the chassis frame.

Matrix $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\hat{\mathbf{M}}^{\Sigma}\mathbf{R}_{\mathrm{d}}$ entails a high computational cost due to the accumulated coupling terms. A new way of taking into account the rods' inertia for enhanced efficiency is now explored. The approach is based on a simple yet effective idea: consider the accumulated rod-related SDIFs as known terms and move them to the right-hand side of the equations of motion. In this manner, the mass matrix is void of rod coupling terms, which results in improved performance. After approximating the rod Cartesian accelerations in Eq. (22), we obtain the following expression:

$$\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\ddot{\mathbf{z}} = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\Big[\hat{\mathbf{Q}}^{\Sigma} - \underbrace{\big(\mathbf{P}^{\Sigma} + \mathbf{P}_{\mathrm{rods}}^{\Sigma}\big)}_{\hat{\mathbf{P}}^{\Sigma}}\Big] - \mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}_{\mathrm{rods}}\hat{\dot{\mathbf{Z}}}_{\mathrm{rods}} \tag{25}$$

where $\hat{\mathbf{P}}^{\Sigma}$ are the accumulated VDIFs including the rods' contribution. Vector $\hat{\dot{\mathbf{Z}}}_{\mathrm{rods}}$ is the approximate vector of rod-related Cartesian accelerations. In the vehicle model example, this can be expressed as

$$\begin{aligned}\hat{\dot{\mathbf{Z}}}_{\mathrm{rods}} \equiv \big\{&\mathbf{0}^{\mathrm{T}}, \hat{\dot{\mathbf{Z}}}_2^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, \ldots, \mathbf{0}^{\mathrm{T}}, \hat{\dot{\mathbf{Z}}}_9^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, \ldots, \mathbf{0}^{\mathrm{T}}, \hat{\dot{\mathbf{Z}}}_{15}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}},\\ &\mathbf{0}^{\mathrm{T}}, \hat{\dot{\mathbf{Z}}}_{18}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, \ldots, \mathbf{0}^{\mathrm{T}}, \hat{\dot{\mathbf{Z}}}_{24}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, \ldots, \mathbf{0}^{\mathrm{T}}, \hat{\dot{\mathbf{Z}}}_{28}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}, \ldots\big\}^{\mathrm{T}}.\end{aligned} \tag{26}$$

Let us now look at the closed-loop equations, derived by applying the second velocity transformation ($\mathbf{R}_{\mathrm{z}}$) and including the approximation of rod-related SDIFs:

$$\mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}}\ddot{\mathbf{z}}^{\mathrm{i}} = \mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\big(\hat{\mathbf{Q}}^{\Sigma} - \hat{\mathbf{P}}^{\Sigma} - \mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\dot{\mathbf{R}}_{\mathrm{z}}\dot{\mathbf{z}}^{\mathrm{i}}\big) - \mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}_{\mathrm{rods}}\hat{\dot{\mathbf{Z}}}_{\mathrm{rods}} \tag{27}$$

or, more compactly,

$$\mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}}\ddot{\mathbf{z}}^{\mathrm{i}} = \mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\big(\hat{\mathbf{Q}}^{\Sigma} - \hat{\mathbf{P}}^{\Sigma} - \mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\dot{\mathbf{R}}_{\mathrm{z}}\dot{\mathbf{z}}^{\mathrm{i}} - \tilde{\mathbf{F}}_{\mathrm{rods}}^{\Sigma}\big). \tag{28}$$

By introducing Eq. (16) and rearranging, Eq. (28) can be rewritten as

$$\mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}}\ddot{\mathbf{z}}^{\mathrm{i}} = \mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left[\hat{\mathbf{Q}}^{\Sigma} - \mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\frac{d(\mathbf{T}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}})}{dt}\dot{\mathbf{z}}^{\mathrm{i}} - \underbrace{\big(\mathbf{P}_{\mathrm{rods}}^{\Sigma} + \tilde{\mathbf{F}}_{\mathrm{rods}}^{\Sigma}\big)}_{\mathbf{P}'^{\Sigma}_{\mathrm{rods}}}\right] \tag{29}$$

which is the final form of the semi-recursive formulation following the rod-related SDIFs approximation approach. We can see, by comparing the left-hand side of Eqs. (24) and (29), that the approximation leads to a sparser mass matrix that has a narrower bandwidth. This

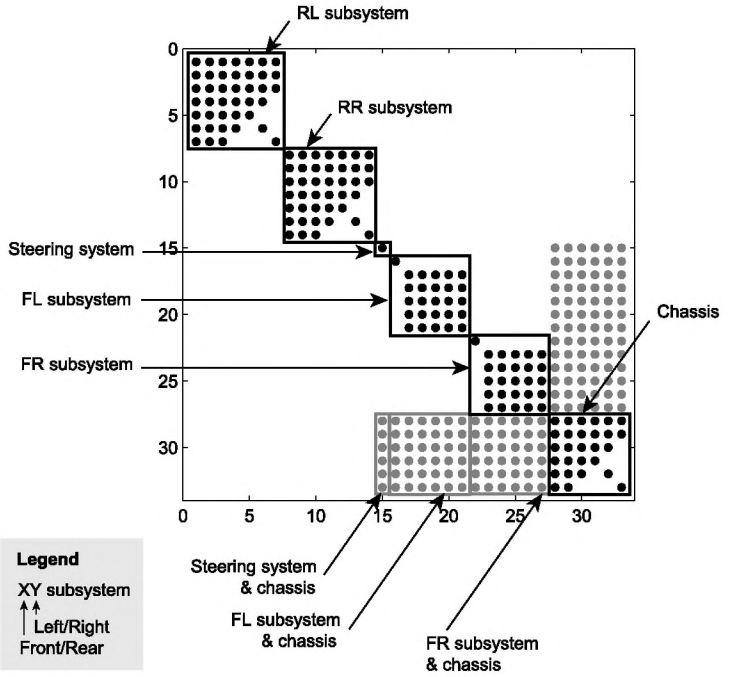

Fig. 5 Structure of matrix $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}$ with approximated rod-related SDIF terms

means that the number of fills and operations in the factorization and solve steps associated with Eq. (29) will be less. The structure of matrix $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}$ is shown in Fig. 5. The diagonal submatrices and the remaining coupling submatrices have the same form as those in Fig. 4. We can extract some additional conclusions by comparing Figs. 4 and 5:

- The chassis is still coupled with the front suspension and steering systems.
- The rear suspension systems and the chassis are no longer coupled.
- The steering system and the front suspension systems are no longer coupled either, since the steering rods connected to the front suspension systems are eliminated.
- The inertia and coupling matrices of the suspension and steering systems and the chassis are simpler than their Fig. 4 counterparts, since the contribution of rods to inertia matrices is eliminated.
- The eliminated coupling matrices and inertia matrices are those shown in Table 3.

A preliminary benefit of this approach is that, due to the sparser structure of matrix $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}$, the numerical integration can be carried out more efficiently [14]. However, we still have to address the evaluation of rod-related SDIF terms. One way of computing such terms is to extrapolate the accelerations based on previous (known) relative accelerations. A number of extrapolation methods are now presented.

The Cartesian accelerations and accumulated rod-related SDIFs shown in Eq. (29) in the current integration time, t, can be expressed as:

$$\hat{\ddot{\mathbf{z}}}_t = \mathbf{f}(\ddot{\mathbf{z}}_{t-h}, \ddot{\mathbf{z}}_{t-2h}, \ldots), \tag{30}$$

$$\hat{\dot{\mathbf{Z}}}_t = \hat{\dot{\mathbf{Z}}}_{i-1} + \mathbf{b}_i \hat{\ddot{z}}_i + \mathbf{d}_i, \tag{31}$$

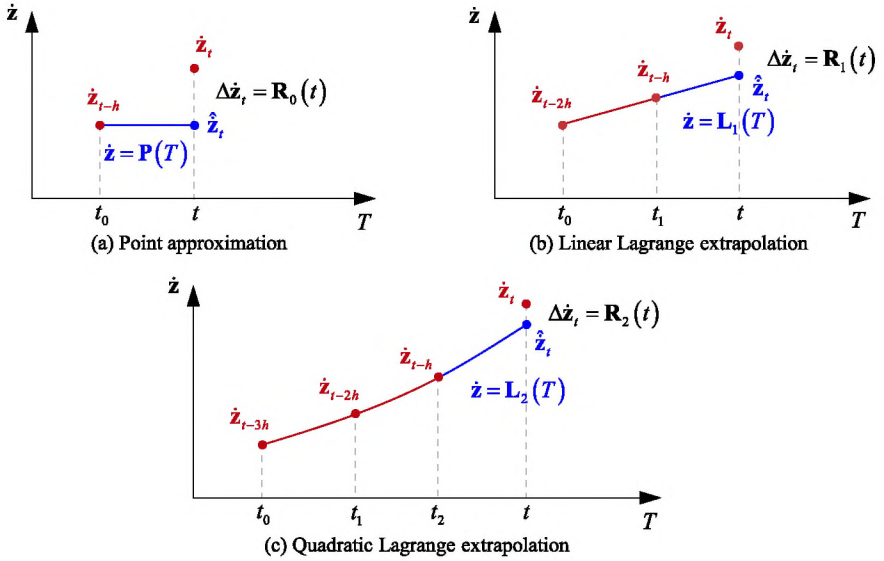

Fig. 6 Geometric description of extrapolation methods

$$\tilde{\mathbf{F}}^{\Sigma}_{\text{rods}} = \mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}_{\text{rods}}\hat{\dot{\mathbf{Z}}}_{t,\text{rods}} \tag{32}$$

where h is the integration time-step and $\mathbf{f}$ is the extrapolation function. Three extrapolation methods of growing complexity have been explored: point approximation and linear and quadratic Lagrange extrapolation methods. These extrapolation methods are depicted in Fig. 6. It can be seen how current relative accelerations are expressed in terms of relative accelerations from previous known integration steps. The curves represent the extrapolation polynomials and the points represent the current and previous acceleration values.

Let us briefly describe the different extrapolation methods [4]. First, the current acceleration, $\hat{\ddot{\mathbf{z}}}_t$, according to the point approximation method (depicted in Fig. 6(a)) can be written as:

$$\hat{\ddot{\mathbf{z}}}_t = \ddot{\mathbf{z}}_{t-h}, \tag{33}$$

$$\mathbf{R}_0(t) = \mathbf{z}^{\text{iii}}_{\xi}\big[t-(t-h)\big] = \mathbf{z}^{\text{iii}}_{\xi}h = \mathrm{O}(h), \quad \xi \in (t-h, t) \tag{34}$$

where $\mathbf{R}_0(t)$ is the corresponding remainder and Roman numerals indicate derivatives of order higher than two.

Second, the generic polynomial expression of the linear Lagrange extrapolation method (shown in Fig. 6(b)) can be expressed as

$$\mathbf{L}_1(T) = \frac{T-t_1}{t_0-t_1}\ddot{\mathbf{z}}_0 + \frac{T-t_0}{t_1-t_0}\ddot{\mathbf{z}}_1. \tag{35}$$

Substituting for time t and time-step h, we obtain the acceleration and remainder:

$$\hat{\ddot{\mathbf{z}}}_t = \mathbf{L}_1(t) = 2\ddot{\mathbf{z}}_{t-h} - \ddot{\mathbf{z}}_{t-2h}, \tag{36}$$

$$\mathbf{R}_1(t) = \mathbf{z}^{\text{iv}}_{\xi}\big[t-(t-h)\big]\big[t-(t-2h)\big] = 2\mathbf{z}^{\text{iv}}_{\xi}h^2 = \mathrm{O}\big(h^2\big), \quad \xi \in (t-2h, t). \tag{37}$$

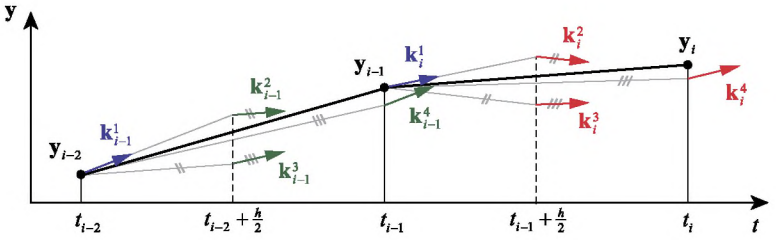

Fig. 7 Geometric description of 4th-order Runge–Kutta integrator

Third, the quadratic Lagrange extrapolation method (see Fig. 6(c)) reads

$$\mathbf{L}_2(T) = \frac{(T-t_1)(T-t_2)}{(t_0-t_1)(t_0-t_2)}\ddot{\mathbf{z}}_0 + \frac{(T-t_0)(T-t_2)}{(t_1-t_0)(t_1-t_2)}\ddot{\mathbf{z}}_1 + \frac{(T-t_0)(T-t_1)}{(t_2-t_0)(t_2-t_1)}\ddot{\mathbf{z}}_2. \tag{38}$$

The acceleration and corresponding remainder can be obtained as:

$$\hat{\ddot{\mathbf{z}}}_t = \mathbf{L}_2(t) = 3\ddot{\mathbf{z}}_{t-h} - 3\ddot{\mathbf{z}}_{t-2h} + \ddot{\mathbf{z}}_{t-3h}, \tag{39}$$

$$\mathbf{R}_2(t) = \mathbf{z}^{\mathrm{v}}_{\xi}\big[t-(t-h)\big]\big[t-(t-2h)\big]\big[t-(t-3h)\big] = 6\mathbf{z}^{\mathrm{v}}_{\xi}h^3 = \mathrm{O}\big(h^3\big),$$

$$\xi \in (t-3h, t). \tag{40}$$

According to the polynomial and remainder expressions, the quadratic Lagrange extrapolation method is the most accurate, whereas the point approximation method is the most efficient.

The way the extrapolation strategy for relative rod accelerations is implemented depends on the specific time integration scheme under consideration. By defining the state vector as $\mathbf{y}^{\mathrm{T}} = \{\mathbf{z}^{\mathrm{T}}, \dot{\mathbf{z}}^{\mathrm{iT}}\}$ according to Maggi's approach [12], we can rewrite the final ODE system (29) in a generic way:

$$\dot{\mathbf{y}} \equiv \begin{Bmatrix} \dot{\mathbf{z}} \\ \ddot{\mathbf{z}}^{\mathrm{i}} \end{Bmatrix} = \mathbf{K}\left(t, \hat{\ddot{\mathbf{z}}}, \begin{Bmatrix} \mathbf{z} \\ \dot{\mathbf{z}}^{\mathrm{i}} \end{Bmatrix}\right) \equiv \mathbf{K}(t, \hat{\ddot{\mathbf{z}}}, \mathbf{y}) \tag{41}$$

which can be solved using standard integrators. Let us consider, as an example, a fixed-step, explicit, 4th-order Runge–Kutta scheme, which we will use in what follows:

$$\mathbf{y}_i = \mathbf{y}_{i-1} + \tfrac{h}{6}\big(\mathbf{k}^1_i + 2\mathbf{k}^2_i + 2\mathbf{k}^3_i + \mathbf{k}^4_i\big), \tag{42}$$

$$\mathbf{k}^1_i \equiv \mathbf{K}(t_{i-1}, \mathbf{y}_{i-1}), \tag{43}$$

$$\mathbf{k}^2_i \equiv \mathbf{K}\big(t_{i-1} + \tfrac{h}{2}, \mathbf{y}_{i-1} + \tfrac{h}{2}\mathbf{k}^1_i\big), \tag{44}$$

$$\mathbf{k}^3_i \equiv \mathbf{K}\big(t_{i-1} + \tfrac{h}{2}, \mathbf{y}_{i-1} + \tfrac{h}{2}\mathbf{k}^2_i\big), \tag{45}$$

$$\mathbf{k}^4_i \equiv \mathbf{K}\big(t_{i-1} + h, \mathbf{y}_{i-1} + h\mathbf{k}^3_i\big). \tag{46}$$

The geometric description of this scheme is shown in Fig. 7. At time-step i, each of the state vector derivative evaluations ($\mathbf{k}^1_i$, $\mathbf{k}^2_i$, $\mathbf{k}^3_i$ and $\mathbf{k}^4_i$) requires the extrapolation of the relative rod accelerations based on previous time points. For obvious reasons, the relative rod accelerations that are closest in time should be used. This means that in order to calculate $\mathbf{k}^1_i$ we can extrapolate based on accelerations up to $\mathbf{k}^1_{i-1}$, whereas in order to calculate $\mathbf{k}^2_i$, $\mathbf{k}^3_i$

and $\mathbf{k}_i^4$ we can extrapolate based on accelerations up to $\mathbf{k}_i^1$. Note that only true accelerations ($\mathbf{k}^1$) can be used in the extrapolation, since intermediate accelerations $\mathbf{k}^2$, $\mathbf{k}^3$ and $\mathbf{k}^4$ are based on non-definitive integration points.

5 Partitioning of rod mass matrix

We have presented how the rod-related SDIF terms can be approximated for faster dynamic simulations. In systems that are heavily dependent on rods such as the sedan car previously analyzed, the loss of precision associated with rod-related SDIF approximation can lead to numerical inaccuracies that are higher than desired. In this section, we partition the rod mass matrix in order to reduce rod-related SDIF errors. This way, the solution accuracy can be improved while preserving computational efficiency.

The last term in the right-hand side of Eq. (25), that is, the vector of accumulated approximate forces, contains the rod mass matrix. This matrix, in turn, can be regarded as the addition of diagonal and coupling terms:

$$\mathbf{R}_\mathrm{d}^\mathrm{T}\mathbf{T}^\mathrm{T}\bar{\mathbf{M}}_\mathrm{rods}\hat{\dot{\mathbf{Z}}} = \mathbf{R}_\mathrm{d}^\mathrm{T}\mathbf{T}^\mathrm{T}\bar{\mathbf{M}}_\mathrm{rods}^\mathrm{d}\hat{\dot{\mathbf{Z}}} + \underbrace{\mathbf{R}_\mathrm{d}^\mathrm{T}\mathbf{T}^\mathrm{T}\bar{\mathbf{M}}_\mathrm{rods}^\mathrm{c}\hat{\mathbf{Z}}_\mathrm{rods}}_{\breve{\mathbf{F}}_\mathrm{rods}^\Sigma} \tag{47}$$

where $\bar{\mathbf{M}}_\mathrm{rods}^\mathrm{d}$ and $\bar{\mathbf{M}}_\mathrm{rods}^\mathrm{c}$ respectively are the diagonal and coupling terms of the rod mass matrix, which can be easily calculated. Vector $\breve{\mathbf{F}}_\mathrm{rods}^\Sigma$ contains the coupling part of the approximate rod-related SDIFs.

Rearranging and taking into account Eq. (15), we can derive the following set of dynamic equations:

$$\begin{aligned}&\mathbf{R}_\mathrm{z}^\mathrm{T}\mathbf{R}_\mathrm{d}^\mathrm{T}\mathbf{M}^\Sigma\mathbf{R}_\mathrm{d}\mathbf{R}_\mathrm{z}\ddot{\mathbf{z}}^\mathrm{i} + \mathbf{R}_\mathrm{z}^\mathrm{T}\mathbf{R}_\mathrm{d}^\mathrm{T}\mathbf{M}_\mathrm{rods}^{\mathrm{d}\Sigma}\left(\mathbf{R}_\mathrm{d}\mathbf{R}_\mathrm{z}\ddot{\mathbf{z}}^\mathrm{i} + \mathbf{R}_\mathrm{d}\dot{\mathbf{R}}_\mathrm{z}\dot{\mathbf{z}}^\mathrm{i} + \dot{\mathbf{R}}_\mathrm{d}\mathbf{R}_\mathrm{z}\dot{\mathbf{z}}^\mathrm{i}\right)\\ &\quad = \mathbf{R}_\mathrm{z}^\mathrm{T}\mathbf{R}_\mathrm{d}^\mathrm{T}\left(\hat{\mathbf{Q}}^\Sigma - \hat{\mathbf{P}}^\Sigma\right) - \mathbf{R}_\mathrm{z}^\mathrm{T}\mathbf{R}_\mathrm{d}^\mathrm{T}\breve{\mathbf{F}}_\mathrm{rods}^\Sigma\end{aligned} \tag{48}$$

where $\mathbf{M}_\mathrm{rods}^{\mathrm{d}\Sigma} \equiv \mathbf{T}^\mathrm{T}\bar{\mathbf{M}}_\mathrm{rods}^\mathrm{d}\mathbf{T}$ is the diagonal component of the accumulated inertia matrix of the closed-loop system.

By introducing Eq. (16) and rearranging, we can rewrite Eq. (48) as

$$\begin{aligned}&\mathbf{R}_\mathrm{z}^\mathrm{T}\mathbf{R}_\mathrm{d}^\mathrm{T}\underbrace{\left(\mathbf{M}^\Sigma + \mathbf{M}_\mathrm{rods}^{\mathrm{d}\Sigma}\right)}_{\hat{\mathbf{M}}_\mathrm{d}^\Sigma}\mathbf{R}_\mathrm{d}\mathbf{R}_\mathrm{z}\ddot{\mathbf{z}}^\mathrm{i}\\ &\quad = \mathbf{R}_\mathrm{z}^\mathrm{T}\mathbf{R}_\mathrm{d}^\mathrm{T}\left[\hat{\mathbf{Q}}^\Sigma - \mathbf{T}^\mathrm{T}\underbrace{\left(\bar{\mathbf{M}} + \bar{\mathbf{M}}_\mathrm{rods}^\mathrm{d}\right)}_{\bar{\mathbf{M}}^\mathrm{d}}\frac{d(\mathbf{T}\mathbf{R}_\mathrm{d}\mathbf{R}_\mathrm{z})}{dt}\dot{\mathbf{z}}^\mathrm{i} - \underbrace{\left(\mathbf{P}_\mathrm{rods}^\Sigma + \breve{\mathbf{F}}_\mathrm{rods}^\Sigma\right)}_{\breve{\mathbf{P}}_\mathrm{rods}^\Sigma}\right]\end{aligned} \tag{49}$$

where $\hat{\mathbf{M}}_\mathrm{d}^\Sigma$ and $\bar{\mathbf{M}}^\mathrm{d}$ are the accumulated inertia matrix and the inertia matrix of the whole system, including the diagonal component of the rods' contribution.

Therefore, the approximate accumulated rod-related SDIFs (only containing coupling terms) can be described as

$$\breve{\mathbf{F}}_\mathrm{rods}^\Sigma = \mathbf{T}^\mathrm{T}\bar{\mathbf{M}}_\mathrm{rods}^\mathrm{c}\hat{\dot{\mathbf{Z}}}_\mathrm{rods} \tag{50}$$

which, compared to the original accumulated rod-related SDIFs in Eq. (32), is much smaller, thereby improving accuracy. Meanwhile, the system inertia matrix $\bar{\mathbf{M}}^\mathrm{d}$ in Eq. (49) is still a

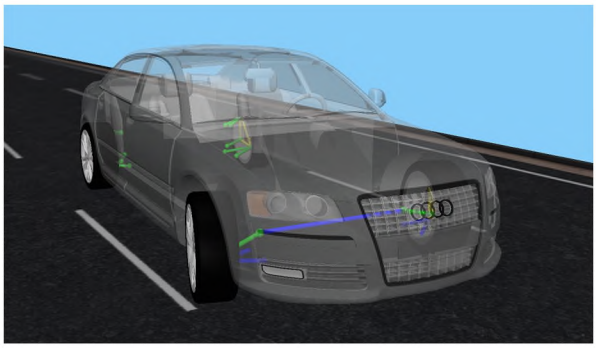

Fig. 8 Multibody model of sedan vehicle

diagonal matrix, which preserves the convenient structure of $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}$ when we replace it by $\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\hat{\mathbf{M}}_{\mathrm{d}}^{\Sigma}\mathbf{R}_{\mathrm{d}}$.

In the vehicle example, the system inertia matrix including the rod mass matrix partition can be written as

$$\bar{\mathbf{M}}^{\mathrm{d}} = \mathrm{diag}\big(\bar{\mathbf{M}}_1, \underline{\bar{\mathbf{M}}_2'}, \bar{\mathbf{M}}_3, \ldots, \bar{\mathbf{M}}_8, \underline{\bar{\mathbf{M}}_9'}, \bar{\mathbf{M}}_{10}, \ldots, \bar{\mathbf{M}}_{14}, \underline{\bar{\mathbf{M}}_{15}'}, \bar{\mathbf{M}}_{16}, \\ \bar{\mathbf{M}}_{17}, \underline{\bar{\mathbf{M}}_{18}'}, \bar{\mathbf{M}}_{19}, \ldots, \bar{\mathbf{M}}_{23}, \underline{\bar{\mathbf{M}}_{24}'}, \bar{\mathbf{M}}_{25}, \ldots, \bar{\mathbf{M}}_{27}, \underline{\bar{\mathbf{M}}_{28}'}, \bar{\mathbf{M}}_{29}, \ldots, \bar{\mathbf{M}}_{33}\big) \tag{51}$$

where $\bar{\mathbf{M}}_i' \equiv \bar{\mathbf{M}}_i + \bar{\mathbf{M}}_{\mathrm{rod},i}^{\mathrm{d}}$. These terms have been underlined for the sake of clarity. They can be computed by assembling the diagonal part of the rods' contribution presented in Table 3 onto the system inertia matrix, $\bar{\mathbf{M}}$.

In short, the partitioning of the rod mass matrix brings about smaller rod-related SDIFs and preserves the structure of the mass matrix, which leads to more accurate results at a low computational cost.

6 Results

In order to verify the effectiveness and efficiency of the approximated rod-related SDIFs with respect to the different extrapolation methods, the 15-DOF sedan vehicle is simulated on a MATLAB/C/C++ environment, as shown in Fig. 8. Tire forces are modeled using Pacejka's Magic Formula. A comparative study has been carried out to assess solution accuracy and computational efficiency. All simulations consist of a slalom test along a flat road and are simulated using a 4th-order Runge–Kutta integrator and an initial velocity of 30 m/s.

6.1 Accuracy

A priori, we know that short time-steps, higher-order extrapolation methods and small rod masses will increase the solution accuracy when the rod-related SDIF approximation is used. The dynamic maneuver has been simulated for one second using the point approximation and linear and quadratic Lagrange extrapolation methods presented earlier. Different time-steps have been employed as well. To assess the accuracy of this method, system-wide relative accelerations are compared to those obtained from the original formulation in Eq. (24). The maximum differences (errors) in relative accelerations are shown in Table 4. The errors

Table 4 Error incurred by rod-related SDIF approximation with different time-steps

h (s)	Error					
	Point approximation		Linear Lagrange		Quadratic Lagrange	
	$\max(\Delta\ddot{\mathbf{z}})$	$\text{norm}(\Delta\ddot{\mathbf{z}})$	$\max(\Delta\ddot{\mathbf{z}})$	$\text{norm}(\Delta\ddot{\mathbf{z}})$	$\max(\Delta\ddot{\mathbf{z}})$	$\text{norm}(\Delta\ddot{\mathbf{z}})$
1×10^{-4}	1.63×10^{-4}	3.50×10^{-4}	5.67×10^{-7}	1.28×10^{-6}	1.68×10^{-8}	3.07×10^{-8}
2×10^{-4}	3.24×10^{-4}	7.00×10^{-3}	2.33×10^{-6}	5.07×10^{-6}	1.11×10^{-7}	1.89×10^{-7}
5×10^{-4}	8.02×10^{-4}	1.70×10^{-3}	1.53×10^{-5}	3.18×10^{-5}	9.85×10^{-7}	2.05×10^{-6}
1×10^{-3}	1.60×10^{-3}	3.40×10^{-3}	6.56×10^{-5}	1.31×10^{-4}	5.29×10^{-6}	1.30×10^{-5}

Table 5 Error incurred by rod-related SDIF approximation with different rod masses

Rod mass	h (s)	Error					
		Point approximation		Linear Lagrange		Quadratic Lagrange	
		$\max(\Delta\ddot{\mathbf{z}})$	$\text{norm}(\Delta\ddot{\mathbf{z}})$	$\max(\Delta\ddot{\mathbf{z}})$	$\text{norm}(\Delta\ddot{\mathbf{z}})$	$\max(\Delta\ddot{\mathbf{z}})$	$\text{norm}(\Delta\ddot{\mathbf{z}})$
0	5×10^{-4}	0	0	0	0	0	0
	1×10^{-3}	0	0	0	0	0	0
$0.5\times$	5×10^{-4}	4.3×10^{-4}	9.2×10^{-4}	7.5×10^{-6}	1.6×10^{-5}	4.8×10^{-7}	9.8×10^{-7}
	1×10^{-3}	8.5×10^{-4}	1.8×10^{-3}	3.1×10^{-5}	6.5×10^{-5}	2.7×10^{-6}	6.6×10^{-6}
$2\times$	5×10^{-4}	1.3×10^{-3}	3.1×10^{-3}	3.4×10^{-5}	6.4×10^{-5}	1.9×10^{-6}	4.0×10^{-6}
	1×10^{-3}	2.6×10^{-3}	6.1×10^{-3}	1.4×10^{-4}	2.6×10^{-4}	8.8×10^{-6}	2.3×10^{-5}
$3\times$	5×10^{-4}	3.0×10^{-3}	5.4×10^{-3}	1.1×10^{-4}	2.2×10^{-4}	1.7×10^{-5}	2.5×10^{-5}
	1×10^{-3}	3.6×10^{-3}	8.4×10^{-3}	2.3×10^{-4}	3.8×10^{-4}	1.5×10^{-5}	3.5×10^{-5}

increase with longer time-steps, and the largest error usually occurs at the end of the simulation because the differences build up over time. Also, the quadratic Lagrange extrapolation method provides the most accurate solutions.

The rod mass also affects the solution accuracy through the inertia matrix, as seen in Eqs. (32) and (50). Thus, we also investigate the accuracy with different rod mass values in order to assess the effectiveness of rod-related SDIFs approximation method. Table 5 shows the error when considering null, half, double and triple rod masses, respectively. Theoretically, the differences between the approximation method and the original formulation should be proportional to the rods' mass and null for the null rod mass case. The results in Table 5 corroborate this and show that the errors are very small even with artificially heavy rods. Linear and quadratic extrapolation methods are much more accurate than the point approximation case. The error, as expected, increases with longer time-steps and builds up over time.

6.2 Efficiency

The dynamic simulation has been timed for each of the approximation methods. All simulations have been run on an Intel® Xeon® machine with a 3.0 GHz CPU and 4 GB of RAM. Table 6 shows the elapsed times for the different rod-related SDIF approximation techniques, as well as the original semi-recursive formulation.

Table 6 Computational efficiency of rod-related SDIF approximation in 10-s and 20-s simulations

h (s)	Formulation	Loop count	Elapsed time (s)	Improvement
5×10^{-4}	Original formulation	80 000	11.577	
	Point approximation		10.403	10.14 %
	Linear Lagrange extrapolation		10.444	9.78 %
	Quadratic Lagrange extrapolation		10.515	9.17 %
1×10^{-3}	Original formulation	80 000	11.586	
	Point approximation		10.376	10.44 %
	Linear Lagrange extrapolation		10.424	10.03 %
	Quadratic Lagrange extrapolation		10.447	9.83 %

The computational savings take place in the state vector derivative function, which is the one directly affected by rod-related SDIF approximations. When using a 0.5-ms integration time-step, the point, linear and quadratic Lagrange approximation methods improve the computational efficiency by 10.14 %, 9.78 % and 9.17 %, respectively. In the 1-ms time-step case, the computational efficiency is improved by 10.44 %, 10.03 % and 9.83 %. The quadratic Lagrange extrapolation method yields the most accurate solutions.

7 Conclusions

We have presented an enhancement to the double-step, semi-recursive formulation developed in [10]. First, the topology of a 15-DOF vehicle model has been fully described and modeled through the aforementioned formulation, including the cut-joint and rod-removal techniques. Second, an original method for approximating rod-related, second-derivative-based inertia forces (SDIF) has been presented, which has improved computational efficiency. Three extrapolation methods have been introduced as part of the rod-related SDIF approximation: point approximation, and linear and quadratic Lagrange extrapolation. Third, the mass matrix of the rods has been partitioned into diagonal and off-diagonal elements to ameliorate the accuracy of the rod-related SDIF approximation. The efficacy and efficiency of the presented approach has been assessed through a series of dynamic simulations where the extrapolation method, the rod mass and the time-step have been varied. An efficiency gain of about 10 % has been observed, along with a satisfactory accuracy. Overall, a very simple, yet effective approximation technique has been presented in the context of efficient formulations for rod modeling in vehicle systems. Systems with more rods or systems depending more heavily on rod inertial forces can greatly benefit from this technique.

Acknowledgements Financial support of the first author from the CSC research fellowship is acknowledged, as well as funding from the Ministry of Science and Innovation of Spain under Research Projects OPTIVIRTEST (TRA2009-14513-C02-01) and DOPTCARR (TRA2012-38826-C02-01).

Appendix: Vehicle model details

Some additional details about the vehicle model depicted in Figs. 1 and 8 are gathered here. The number of bodies, joints, constraints and coordinates is shown in Table 7. The DOF count is detailed in Table 8. Finally, detailed descriptions of rod elements and cut joints are presented in Tables 9 and 10, respectively.

Table 7 Element count

Item	#
Bodies (including auxiliary bodies)	34
Auxiliary bodies	19
Joints (including cut joints)	35
Relative coordinates	33
Independent relative coordinates	14
Subsystems	5
Driving coordinate	1
Constraint equations	18
Eliminated rods	12
Eliminated joints	2

Table 8 Degree of freedom count

Item	#	DOF/item	Σ DOF
Parts (no ground)	33	6	198
Prismatic joints	12	−5	−60
Revolute joints	21	−5	−105
Spherical joints	2	−3	−6
Removed rods	12	−1	−12
Total DOF			15

Table 9 Rods

ID	Mass (kg)	Connected bodies	Connected points	Subsystem
1	2.0	Stabilization bar & Front right knuckle	$\mathbf{r}_a, \mathbf{r}_b$	Front right subsystem
2	2.0	Stabilization bar & Front left knuckle	$\mathbf{r}_c, \mathbf{r}_d$	Front left subsystem
3	2.0	Chassis frame & Rear right knuckle	$\mathbf{r}_e, \mathbf{r}_f$	Rear right subsystem
4	2.0	Chassis frame & Rear right knuckle	$\mathbf{r}_g, \mathbf{r}_h$	
5	4.0	Chassis frame & Rear right knuckle	$\mathbf{r}_i, \mathbf{r}_j$	
6	1.0	Chassis frame & Rear right knuckle	$\mathbf{r}_k, \mathbf{r}_l$	
7	3.0	Chassis frame & Rear right knuckle	$\mathbf{r}_m, \mathbf{r}_n$	
8	2.0	Chassis frame & Rear left knuckle	$\mathbf{r}_o, \mathbf{r}_p$	Rear left subsystem
9	2.0	Chassis frame & Rear left knuckle	$\mathbf{r}_q, \mathbf{r}_r$	
10	4.0	Chassis frame & Rear left knuckle	$\mathbf{r}_s, \mathbf{r}_t$	
11	1.0	Chassis frame & Rear left knuckle	$\mathbf{r}_u, \mathbf{r}_v$	
12	3.0	Chassis frame & Rear left knuckle	$\mathbf{r}_w, \mathbf{r}_x$	

Table 10 Cut joints

ID	Joint type	Connected bodies	Connected points	Subsystem
I	Spherical	Right control arm & Front right knuckle	$\mathbf{r}'_a, \mathbf{r}'_b$	Front right subsystem
II	Spherical	Left control arm & Front left knuckle	$\mathbf{r}'_c, \mathbf{r}'_d$	Front left subsystem

References

1. Anderson, K.S., Duan, S.: A hybrid parallelizable low-order algorithm for dynamics of multi-rigid-body system. Part I. Chain systems. Math. Comput. Model. **30**(9–10), 193–215 (1999)
2. Anderson, K.S., Duan, S.: Highly parallelizable low-order dynamics simulation algorithm for multi-rigid-body systems. J. Guid. Control Dyn. **23**(2), 355–364 (2000)
3. Bae, D., Lee, J., Cho, H., Yae, H.: An explicit integration method for realtime simulation of multibody vehicle models. Comput. Methods Appl. Mech. Eng. **187**(1–2), 337–350 (2000)
4. Brezinski, C., Zaglia, M.R.: Extrapolation Methods: Theory and Practice. North-Holland, Amsterdam (1991)
5. Cuadrado, J., Cardenal, J., Morer, P., Bayo, E.: Intelligent simulation of multibody dynamics: space-state and descriptor methods in sequential and parallel computing environments. Multibody Syst. Dyn. **4**(1), 55–73 (2000)
6. Cuadrado, J., Dopico, D.: A hybrid global-topological real-time formulation for multibody systems. In: Fourth Symposium on Multibody Dynamics and Vibration, at the ASME Nineteenth Biennial Conference on Mechanical Vibration and Noise. ASME, New York (2003)
7. Cuadrado, J., Dopico, D., Gonzalez, M., Naya, M.A.: A combined penalty and recursive real-time formulation for multibody dynamics. J. Mech. Des. **126**(4), 602–608 (2004)
8. Cuadrado, J., Gutiérrez, R., Naya, M.A., Morer, P.: A comparison in terms of accuracy and efficiency between a MBS dynamic formulation with stress analysis and a non-linear FEA code. Int. J. Numer. Methods Eng. **51**(9), 1033–1052 (2001)
9. García de Jalón, J.: Twenty-five years of natural coordinates. Multibody Syst. Dyn. **18**(1), 15–33 (2007)
10. García de Jalón, J., Álvarez, E., de Ribera, F.A., Rodríguez, I., Funes, F.J.: A fast and simple semi-recursive formulation for multi-rigid-body systems. In: Advances in Computational Multibody Systems. Computational Methods in Applied Sciences, vol. 2, pp. 1–23 (2005)
11. García de Jalón, J., Callejo, A.: A straight methodology to include multibody dynamics in graduate and undergraduate subjects. Mech. Mach. Theory **46**(2), 168–182 (2011)
12. García de Jalón, J., Callejo, A., Hidalgo, A.F.: Efficient solution of Maggi's equations. J. Comput. Nonlinear Dyn. **7**(2), 021,003 (2012)
13. García de Jalón, J., Unda, J., Avello, A.: Natural coordinates for the computer analysis of multibody systems. Comput. Methods Appl. Mech. Eng. **56**(3), 309–327 (1986)
14. Hidalgo, A.F., García de Jalón, J.: Real-time dynamic simulations of large road vehicles using dense, sparse, and parallelization techniques. J. Comput. Nonlinear Dyn. **10**(3), 031,005 (2015)
15. Jerkovsky, W.: The structure of multibody dynamic equations. J. Guid. Control Dyn. **1**(3), 173–182 (1978)
16. Kim, S., Vanderploeg, M.: A general and efficient method for dynamic analysis of mechanical system using velocity transformations. J. Mech. Des. **108**(2), 176–182 (1986)
17. Mráz, L., Valásek, M.: Solution of three key problems for massive parallelization of multibody dynamics. Multibody Syst. Dyn. **29**(1), 21–39 (2013)
18. Negrut, D., Serban, R., Mazhar, H., Heyn, T.: Parallel computing in multibody system dynamics: why, when, and how. J. Comput. Nonlinear Dyn. **9**(4), 041,007 (2014)
19. Negrut, D., Serban, R., Potra, F.A.: A topology based approach to exploiting sparsity in multibody dynamics: joint formulation. Mech. Struct. Mach. **25**(2), 221–241 (1997)
20. Rodríguez, J.I., Jiménez, J.M., Funes, F.J., García de Jalón, J.: Recursive and residual algorithms for the efficient numerical integration of multi-body systems. Multibody Syst. Dyn. **11**(4), 295–320 (2004)
21. Saha, S.K., Schiehlen, W.O.: Recursive kinematics and dynamics for closed loop multibody systems. Mech. Struct. Mach. **29**(2), 143–175 (2001)
22. von Schwerin, R.: Multibody System Simulation, Numerical Methods, Algorithms and Software. Springer, Berlin (1999)
23. Serban, R., Haug, E.: Globally independent coordinates for real-time vehicle simulation. J. Mech. Des. **122**, 575–582 (2000)
24. Serban, R., Negrut, D., Haug, E.J., Potra, F.A.: A topology-based approach for exploiting sparsity in multibody dynamics in Cartesian formulation. Mech. Struct. Mach. **25**(3), 379–396 (1997)
25. Shabana, A.A., Wehage, R.A.: A coordinate reduction technique for dynamic analysis of spatial substructures with large angular rotations. J. Struct. Mech. **11**(3), 401–431 (1983)
26. Tsai, F.F., Haug, E.J.: Real-time multibody system dynamic simulation. Part I. A modified recursive formulation and topological analysis. Mech. Struct. Mach. **19**(1), 99–127 (1991)